

alter VBA code to function on selection set

Posted by Christopher Oneal - 2008/06/10 14:13

Hello all.

Does some one have some ideas on how to get the following code (author Miles Hitchen

<http://forums.esri.com/Thread.asp?c=93&f=987&t=206832>)

to work on > 1 selected features? In its current form it does not iterate through a selection, and I am trying to pickup VBA and am, well, challenged here. I have tried a few suggestions with altering pQFilter but no dice.

```
Public Sub GridQuadrilateral()  
Dim pMxDoc As IMxDocument  
Dim pInFtrLyr As IFeatureLayer  
Dim pFtrSel As IFeatureSelection  
Dim pOutFtrLyr As IFeatureLayer  
Dim pPolygon As IPolygon  
Dim pSegColl As ISegmentCollection  
Dim lIdx As Long  
Dim lCornerIdx(3, 1) As Double  
Dim l As Long  
  
    ' Get the first selected polygon on the first layer  
    Set pMxDoc = ThisDocument  
    Set pInFtrLyr = pMxDoc.FocusMap.Layer(0)  
    Set pFtrSel = pInFtrLyr  
    Set pPolygon = pInFtrLyr.FeatureClass.GetFeature(pFtrSel.SelectionSet.IDs.Next).Shape  
    Set pSegColl = pPolygon  
  
    ' Get the corner coords of the quad  
    lIdx = 0  
    For l = 0 To 3  
        lIdx = GetIndexOfNextCornerSegment(lIdx, pPolygon)  
        lCornerIdx(l, 0) = pSegColl.Segment(lIdx).FromPoint.X  
        lCornerIdx(l, 1) = pSegColl.Segment(lIdx).FromPoint.y  
    Next l  
  
    ' Calculate the grid coordinates and store in array  
    Const GRID_SIZE = 4  
    Dim dStep As Double  
    Dim i As Long, j As Long  
    Dim dx As Double, dy As Double  
    Dim pt1x As Double, pt1y As Double, pt2x As Double, pt2y As Double  
    Dim nx As Double, ny As Double  
    Dim dCoords(GRID_SIZE, GRID_SIZE, 1) As Double  
  
    dStep = 1 / CDBl(GRID_SIZE)  
  
    For i = 0 To GRID_SIZE  
        dx = (lCornerIdx(1, 0) - lCornerIdx(0, 0)) * dStep * i  
        dy = (lCornerIdx(1, 1) - lCornerIdx(0, 1)) * dStep * i  
        pt1x = lCornerIdx(0, 0) + dx  
        pt1y = lCornerIdx(0, 1) + dy  
  
        dx = (lCornerIdx(2, 0) - lCornerIdx(3, 0)) * dStep * i  
        dy = (lCornerIdx(2, 1) - lCornerIdx(3, 1)) * dStep * i  
        pt2x = lCornerIdx(3, 0) + dx  
        pt2y = lCornerIdx(3, 1) + dy  
  
        For j = 0 To GRID_SIZE  
            dx = (pt2x - pt1x) * dStep * j  
            dy = (pt2y - pt1y) * dStep * j  
            nx = pt1x + dx  
            ny = pt1y + dy  
  
            dCoords(j, i, 0) = nx
```

```

        dCoords(j, i, 1) = ny

    Next j
Next i

' Now create the polygons
Dim pFtrCls As IFeatureClass
Dim pFtrCsr As IFeatureCursor
Dim pFtrBfr As IFeatureBuffer
Dim pPtColl As IPointCollection
Dim pPt As IPoint

Set pOutFtrLyr = pMxDoc.FocusMap.Layer(1)
Set pFtrCls = pOutFtrLyr.FeatureClass
Set pFtrBfr = pFtrCls.CreateFeatureBuffer
Set pFtrCsr = pFtrCls.Insert(True)

For i = 0 To GRID_SIZE - 1
    For j = 0 To GRID_SIZE - 1

        Set pPtColl = New Polygon
        Set pPt = New point
        pPt.PutCoords dCoords(j, i, 0), dCoords(j, i, 1)
        pPtColl.AddPoint pPt
        pPt.PutCoords dCoords(j, i + 1, 0), dCoords(j, i + 1, 1)
        pPtColl.AddPoint pPt
        pPt.PutCoords dCoords(j + 1, i + 1, 0), dCoords(j + 1, i + 1, 1)
        pPtColl.AddPoint pPt
        pPt.PutCoords dCoords(j + 1, i, 0), dCoords(j + 1, i, 1)
        pPtColl.AddPoint pPt
        pPt.PutCoords dCoords(j, i, 0), dCoords(j, i, 1)
        pPtColl.AddPoint pPt

        Set pFtrBfr.Shape = pPtColl
        pFtrCsr.InsertFeature pFtrBfr

    Next j
Next i

pMxDoc.ActiveView.Refresh

End Sub

Private Function GetIndexOfNextCornerSegment(IStartIdx As Long, pPolygon As IPolygon) As Long
Dim PI As Double
Dim pSegColl As ISegmentCollection
Dim pLine1 As ILine, pLine2 As ILine
Dim I As Long
Dim INxtIdx As Long
Dim dAng As Double

PI = Atn(1) * 4
Set pSegColl = pPolygon
For I = 0 To pSegColl.SegmentCount - 2
    INxtIdx = IStartIdx + I
    If INxtIdx = pSegColl.SegmentCount Then INxtIdx = 0
    Set pLine1 = pSegColl.Segment(INxtIdx)
    INxtIdx = INxtIdx + 1
    If INxtIdx = pSegColl.SegmentCount Then INxtIdx = 0
    Set pLine2 = pSegColl.Segment(INxtIdx)
    dAng = Abs(pLine1.Angle - pLine2.Angle) * 180 / PI
    If dAng > 20 Then
        ' The start point of this segment is a corner point
    
```

```
        GetIndexofNextCornerSegment = INxtIdx
    Exit Function
End If
Next I
```

```
GetIndexofNextCornerSegment = -1
End Function
```

Re:alter VBA code to function on selection set

Posted by Scott Davis - 2008/06/10 16:20

What a cool tool! Give this a try and let me know how it turns out:

```
Public Sub GridQuadrilateral()
    Dim pMxDoc As IMxDocument
    Dim pInFtrLyr As IFeatureLayer
    Dim pFtrSel As IFeatureSelection
    Dim pOutFtrLyr As IFeatureLayer
    Dim pPolygon As IPolygon
    Dim pSegColl As ISegmentCollection
    Dim lIdx As Long
    Dim lCornerIdx(3, 1) As Double
    Dim l As Long
    Dim pSelectionSet As ISelectionSet

    ' Get the selection set of the first layer in TOC
    Set pMxDoc = ThisDocument
    Set pInFtrLyr = pMxDoc.FocusMap.Layer(0)
    Set pFtrSel = pInFtrLyr
    Set pSelectionSet = pFtrSel.SelectionSet

    ' get cursor of selection set
    Dim pFCursor As IFeatureCursor
    pSelectionSet.Search Nothing, True, pFCursor

    ' get first feature
    Dim pFeature As IFeature
    Set pFeature = pFCursor.NextFeature()

    ' loop through all selected polygons
    Do Until pFeature Is Nothing

        ' get polygon
        Set pPolygon = pFeature.Shape
        Set pSegColl = pPolygon

        ' Get the corner coords of the quad
        lIdx = 0
        For l = 0 To 3
            lIdx = GetIndexofNextCornerSegment(lIdx, pPolygon)
            lCornerIdx(l, 0) = pSegColl.Segment(lIdx).FromPoint.X
            lCornerIdx(l, 1) = pSegColl.Segment(lIdx).FromPoint.Y
        Next l

        ' Calculate the grid coordinates and store in array
        Const GRID_SIZE = 4
        Dim dStep As Double
        Dim i As Long, j As Long
        Dim dx As Double, dy As Double
        Dim pt1x As Double, pt1y As Double, pt2x As Double, pt2y As Double
```

```
Dim nx As Double, ny As Double
Dim dCoords(GRID_SIZE, GRID_SIZE, 1) As Double
```

```
dStep = 1 / CDBl(GRID_SIZE)
```

```
For i = 0 To GRID_SIZE
    dx = (ICornerIdx(1, 0) - ICornerIdx(0, 0)) * dStep * i
    dy = (ICornerIdx(1, 1) - ICornerIdx(0, 1)) * dStep * i
    pt1x = ICornerIdx(0, 0) + dx
    pt1y = ICornerIdx(0, 1) + dy
```

```
    dx = (ICornerIdx(2, 0) - ICornerIdx(3, 0)) * dStep * i
    dy = (ICornerIdx(2, 1) - ICornerIdx(3, 1)) * dStep * i
    pt2x = ICornerIdx(3, 0) + dx
    pt2y = ICornerIdx(3, 1) + dy
```

```
    For j = 0 To GRID_SIZE
        dx = (pt2x - pt1x) * dStep * j
        dy = (pt2y - pt1y) * dStep * j
        nx = pt1x + dx
        ny = pt1y + dy
```

```
        dCoords(j, i, 0) = nx
        dCoords(j, i, 1) = ny
```

```
    Next j
Next i
```

```
' Now create the polygons
Dim pFtrCls As IFeatureClass
Dim pFtrCsr As IFeatureCursor
Dim pFtrBfr As IFeatureBuffer
Dim pPtColl As IPointCollection
Dim pPt As IPoint
```

```
Set pOutFtrLyr = pMxDoc.FocusMap.Layer(1)
Set pFtrCls = pOutFtrLyr.FeatureClass
Set pFtrBfr = pFtrCls.CreateFeatureBuffer
Set pFtrCsr = pFtrCls.Insert(True)
```

```
For i = 0 To GRID_SIZE - 1
    For j = 0 To GRID_SIZE - 1
```

```
        Set pPtColl = New Polygon
        Set pPt = New Point
        pPt.PutCoords dCoords(j, i, 0), dCoords(j, i, 1)
        pPtColl.AddPoint pPt
        pPt.PutCoords dCoords(j, i + 1, 0), dCoords(j, i + 1, 1)
        pPtColl.AddPoint pPt
        pPt.PutCoords dCoords(j + 1, i + 1, 0), dCoords(j + 1, i + 1, 1)
        pPtColl.AddPoint pPt
        pPt.PutCoords dCoords(j + 1, i, 0), dCoords(j + 1, i, 1)
        pPtColl.AddPoint pPt
        pPt.PutCoords dCoords(j, i, 0), dCoords(j, i, 1)
        pPtColl.AddPoint pPt
```

```
        Set pFtrBfr.Shape = pPtColl
        pFtrCsr.InsertFeature pFtrBfr
```

```
    Next j
Next i
```

```
' advance to next polygon in selection set
Set pFeature = pFCursor.NextFeature()
```

Loop

pMxDoc.ActiveView.Refresh

End Sub

=====

Re:alter VBA code to function on selection set

Posted by Christopher Oneal - 2008/06/13 11:34

That is just beautiful. Thanks so much for your the help. The structure is more similar to python than I had thought it to be. Thanks a million

=====